FAST MICROARRAY EXPRESSION DATA ANALYSIS METHOD FOR NETWORK EXPLORATION

FIELD OF THE INVENTION

This invention relates to the field of bio-informatics and more particularly toward a method and algorithm for network exploration and reconstruction. The invention also has application to information, computer, software and data processing systems and more specifically to a method for facilitating the multiple correlation and comparisons of gene, protein and feature selection data.

BACKGROUND OF THE INVENTION

The micro-array was developed in 1995 to measure gene data in a massively parallel fashion. Since that time a significant increase in the amount of data per experiment has occurred (See http://www-binf.bio.uu.nl/~dutilh/gene-networks/thesis.html). In the case of gene exploration, this extensive data is important for use in assessing genes and their influence on expressed states in organisms. In particular, it is necessary to assess the function and operation of a particular gene; a gene being defined as a prescribed area on a nucleic acid molecule that is necessary for the production of a final expressed phenotype of an individual. On a more complex and broader scale, the interaction network is also of interest due to its influence in regulating higher cellular, physiological and behavioral properties. Recent attempts are being made to reconstruct the precise interaction network or its fragments based on large-scale array experiments for a condition-specific database, e.g., melanoma (Bittner et al., 2000). The critical first step in these efforts is to find the smallest subset of (predictors), within a desirable degree of precision, related to an arbitrary target. Based on such set of predictors, computed for every target of interest, it is possible to find the smallest set that can explain or predict behavior of any target in terms of expression. In the case of genes, finding the smallest set to predict a prescribed behavior could be a very complicated and arduous task given the massive amount of data that results from analyzing a complete organism's genome.

Most important to scientists is the ability to select a minimal cardinality set that can represent a whole set of expressed information. In the pattern recognition literature, this is known as feature selection or dimensionality reduction, depending on the context.

5   The issue at hand now is a question of mathematics and computation rather than pure biology. In particular, the specific problem at focus has been addressed from a computational standpoint. A number of algorithms can be applied from other fields and areas of study to help solve this arduous task. The specific problem at focus, from a computation standpoint, is to find the best (with respect to a given

10   quality of function) k-tuples, from a set of n features, for many values of k. One method to find the best k-tuple predictor subset is to conduct an exhaustive search through all possible k-tuples. Although this approach always leads to the best solution, it becomes intractable for even moderate values of k (the computational time grows exponentially with k).

15   Also important to bio-informatics will be the methods developed for pattern recognition. In the context of pattern recognition, machine learning, data mining and their applications to various subject areas, e.g., medical diagnostics, manufacturing test design, image recognition etc., a similar problem of subset selection, known as feature selection is important. A number of approaches have been proposed and

20   designed to address these problems or issues. The approaches include and are not limited to, sequential (backward and forward) search techniques, floating search techniques and genetic or protein algorithms. However, methods based on the sequential search techniques suffer from the nesting effect, i.e, they overlook good feature sets consisting of individually poor quality features. A second method called

25   the floating search methods (Pudil et al., 2000; Somol et al., 2000) attempt to avoid the nesting problem by successively adding the best and removing the worst subsets of features from the candidate set of features. This introduces an exponential complexity in the search when the size of a subset grows. A significant drawback of these methods is that they become slow for large dimensional data sets as is the case

30   with biological expression data. Genetic or biological algorithms also do not have well defined stopping criteria and, in principle, can be exponentially complex.

Most importantly, the above methods and algorithms are intended to be applied in the field of array data processing to enable computationally efficient

searches for the smallest subset of features that predict a target's expression levels within a given level of confidence or accuracy.

It would, therefore, be desirable to develop a method and algorithm that determines "good" solution sets with high probability in linear time, with respect to total number of features in a predictor set. For this reason, "sequential forward selection"(SFS) (Bishop, 1997; Pudil et al., 2000; Somol & Pudil, 2000) was developed to add the best (the one that leads to the largest improvement in the value of the quality function) new feature, at each successive stage of the algorithm, to the current set of features until the needed number of features is reached. It follows from construction that SFS suffers from the nesting problem and always overlooks better solutions sets whose features are of mediocre or poor quality. This is one of the shortcomings addressed by the present invention. While "sequential floating forward selection" (SFFS) also addresses the nesting problem, it maintains exponential time complexity for large data sets. The second shortcoming that this invention addresses is the exponential time complexity to find "good" solutions. The proposed method and invention finds a "good" solution set with high probability in linear time with respect to number of predictors. One of the floating search algorithms, called "oscillating search", (Somol & Pudil, 2000) can also find approximate solutions in linear time. However, the present invention and method guarantees an equal or better quality solution while maintaining the linear time complexity. In addition, the same generic method or algorithm can be used not only for gene network reconstruction, but also can be applied to protein data, feature selection for classification and other biological data that is very large and complex to organize and analyze.

SUMMARY OF THE INVENTION

The invention is a method for determining a predictor set of features associated with a target. The method comprises the steps of selecting a predictor set of features, adding a complement to the predictor set based on a quality of prediction, checking to see if all of the features of the predictor set are repeated and then removing one feature from the predictor set. The algorithm and method repeats the steps of adding, checking and removing features until the features of the predictor set

are repeated. If the features of the predictor set are repeated, the algorithm and method terminate.

More specifically, the invention is a method for probabilistically determining a subset of features of size k that are closely related to a given target in terms of a

5   selected quality function. The method of the invention operates by allowing a user to select a target of choice and the size (k) of the predictor set. Once a target has been selected, the method starts by selecting an arbitrary (ordered) subset of features of size k-1 and iteratively adds and removes single features (in order) from the selected subset. This process is iterated until a subset of features of size k is found whose

10   quality of prediction of the target can no longer be improved by the process of deletion followed by addition of a feature. The algorithm terminates at this stage. In each iteration, the comparisons are based on a quality function that determines a quality of prediction associated between the predictors and the target. The method of invention can easily be extended to probabilistically determine the smallest (in size)

15   subset of features that are closely related to a given target within an *a priori* set tolerance level in terms of a selected quality function. The method then takes as input a given target (user selected) and iteratively applies the method of invention for subsets of size 1, 2, 3, ..., k, etc., until a predictor set that is closely related to the target expression, within the *a priori* set threshold, is found. The method can also be

20   used for classification of experiments. The method in this case defines a target in terms of a vector of numbers representing the class of experiments under consideration. The method can then be used to identify a subset of features which can classify the data.

25

BRIEF DESCRIPTION OF THE DRAWINGS

Embodiments of the invention will now be described with reference to the drawings in which:

30   FIG. 1 illustrates a schematic view of the present invention in vector format showing the target and the predictors.

FIG. 2 shows a block diagram of the method of the invention.

FIG. 3 shows how the GSSA algorithm makes comparisons of data.

FIG. 4 shows a simulated plot of the number of attractors v. the number of experiments.

FIG. 5 shows a simulated plot of the probability of finding an optimal attractor v. number of genes.

5 FIG. 6 shows a simulated plot of the execution time v. number of genes.

FIG. 7 shows a simulated plot of the execution time v. number of predictors.

FIG. 8 shows a simulated plot of the log of the execution time v. number of genes.

10 **DETAILED DESCRIPTION OF THE INVENTION**

Before describing the present invention in detail, it is to be understood that this invention is not limited to specific compositions, process steps, or equipment, as such may vary. It is also to be understood that the terminology used herein is for the
15 purpose of describing particular embodiments only, and is not intended to be limiting. The invention has broad based use and application in a variety of fields including most importantly the fields of chemistry, biochemistry, computer science and biology.

It must be noted that, as used in this specification and the appended claims, the singular forms "a", "an" and "the" include plural referents unless the context clearly
20 dictates otherwise. Thus, for example, reference to "an attractor" includes more than one attractor, reference to "a predictor" includes a plurality of predictors and the like.

In describing and claiming the present invention, the following terminology will be used in accordance with the definitions set out below.

The term "feature" shall mean expression levels or biological data of a defined
25 gene, protein, or other biological function or component under consideration and over a prescribed number of experiments. It is also the smallest sized component or item that can not be further reduced in a predictor set.

The term "network reconstruction" shall mean the process, apparatus, steps and algorithms used in determining associated and/or non-associated pathways in a
30 data set. In particular, it shall mean the relationships or pathways between two or more predictor sets.

The term "target" shall have a broad based meaning to include, proteins, genes, immunological information, feature selection for classification, and other complex biological and chemical data and or components that may be defined over a

number of experiments. The term has particular meaning to chemical and biochemical data that is extensive, complex and difficult to analyze. For instance, in the case of genes it shall mean the expression levels over the number of experiments of a selected gene of interest.

5   The term "predictor set" shall have a broad based meaning to include, proteins, genes, immunological information, feature selection for classification, and other complex biological and chemical data and or components for a given size k, that are used to compute or predict an associated quality or characteristic of a target. For instance, in the case of gene vectors of a given size, k, it is used to compute or predict

10   an expression level of a target gene vector.

      The term "predictor(s)" is used for a feature that is part of a predictor set.

      The term "prediction" is a vector, computed by using a linear/non-linear function of features in the predictor set (although, we describe the proposed invention in terms of linear prediction function, the method is not limited to the same; it can be

15   extended to other prediction functions such as non-linear, etc.).

      The term "quality" or "quality of prediction" shall herein mean the distance between the predictor and the target. The smaller the distance between the predictor and the target the better the quality of prediction. Geometrically, for k=2, quality is the distance between the target and the plane formed by the two features in the

20   predictor set. It should be noted that this definition of the quality function should not be interpreted as limiting and any other computable function may be used.

      An "attractor" shall mean a set of a given size k of features such that the quality can't be improved by replacing one feature in the set. In other words, in the case of gene data, a set of genes S is an attractor if the quality of its prediction of the

25   target gene, G, cannot be improved by substituting only one gene in this set (in some sense an attractor is a local minima). It should be noted that the best solution is always an attractor.

      The term "complement" shall mean a feature (when looking for a predictor set size of k features it is defined as follows: a feature g is called a complement to a given

30   set of k-1 features if no other feature, along with this given set of k-1 features, can form a higher quality set of k predictors). In the case k=2, gene g* is called a complement to feature g if the "quality" $Q(.,.)$ of the couple (g,g*) is no worse than that of any couple (g,h); $Q(g,g^*) \leq Q(g,h)$ for any h.

The term "k-tuples" shall mean a group of size k. For instance, if k=2, then we call them a "couple". In addition, if k=3 we call them a "triplet". This is an abbreviated term to show the relationship between group size and designated pairings.

The term "good solution" shall mean a set of predictors having a given size with high enough quality.

The term "M-dimensional space" shall mean a variety of orientations and positions in space (i.e, M=1... 1000, arbitrary and large).

When a search for the best (in quality) group of individuals is conducted, the term "nesting" or "nesting effect" shall mean procedures that are based on the assumption that a good in quality group consists of good in quality subgroups, overlooking solutions made up of mediocre or poor (in quality) individuals.

The term "clustering" or "cluster" shall mean associations made between previously non-associated groups or features. Clusters are based upon potential pathway interactions rather than just upon similar expression activity. Use of the method or technique in this way can also allow the identification of genes, proteins or similar type molecules, which are involved in predictor sets for many targets, again leading to pathway identification.

The array is a significant new technology aimed at providing a top down picture of the intimate processes of an organism. Whether an array is implemented using photolithography and silicon-based fabrication, capillary printing heads on a glass slide, or ink-jet technology, it allows for quantification of large amounts of data simultaneously. Few years have passed since the first micro-array based biological results were published and it already seems unthinkable to tackle the complexities of the workings of the cell without these devices. However, there remains unsolved image processing as well as computational and mathematical difficulties associated with the extraction and validation of data for micro-array assays.

Network reconstruction's main function is to discover the existence and determine the association of multiple dependencies between biological or transcriptional data that leads to the identification of possible associated pathways. There are a number of potential motivations for constructing algorithms, and computer software for network reconstruction. For instance, diagnostics and the diagnostic industry can use these techniques and tools for application in disease identification. Other potential valuable applications include treatment selection and

outcome prediction. In addition, the derived information can be further applied to aid in drug development and areas of feature selection for classification.

The references cited in this application are incorporated in this application by reference. However, cited references are not admitted to be prior art to this

5   application.

Feature Selection in Pattern Recognition

The problem of network reconstruction, based on micro-array data, can be

10   reduced to finding dependencies within a subset of features in terms of their expression levels. One meaningful option to address this problem is to find a set of best k predictors for any target of interest.

In the context of pattern recognition, machine learning, data mining and their applications to various subject areas, e.g., medical diagnostics, manufacturing test

15   design, image recognition, etc., a similar problem of subset selection, known as feature selection is faced. We have discussed this related work in feature selection and the advantages of the proposed method over these works in the Section "Background of the Invention". We concentrate here on the method of invention.

The method described below is somewhat similar to what are called the

20   "sequential forward selection" (SFS) and "sequential floating forward selection" (SFFS) methods. SFS adds the "best" (i.e. the one that leads to the largest improvement in the value of the quality function) new feature, at each successive stage of the algorithm, to the current set of features until the needed number of features is reached. In particular, SFS suffers from the nesting problem and always

25   overlooks better solution sets whose features are of mediocre or poor quality. This is one of the shortcomings addressed by this invention. While SFFS also addresses the nesting problem, it maintains exponential time complexity for large data sets. The second shortcoming that this invention addresses is the exponential time complexity to find "good" solutions. The method finds a "good" solution set in linear (with

30   respect to k) time with high probability. One of the floating search algorithms, called "oscillating search", can also find an approximate solution in linear time. However, the method guarantees an equal or better quality solution while maintaining the linear time complexity. The methods and algorithm of the present invention may be used and employed in a variety of systems that receive or export large volumes of data or

8

information. In particular, the algorithm and/or method have potential application with computer and/or computer software. The algorithm may be used, employed or supplied with other similar hardware, software, or systems that are well known in the art.

5   Referring now to FIG. 1, the first step in a network reconstruction entails the search for strong linear dependencies among associated data, i.e.:

$$G \sim F (g, h. ...) \tag{1}$$

10   where G 120 is the target of interest and g 140, h 150 are predictors. The function F can be linear or non-linear. In this application, we investigate only the linear case:

$$F = \alpha * g + \beta * h \tag{2}$$

15 where α and β are constants.

     The quality of a linear prediction of a target G 120, associated with a set of features, g 140, h 150, is given by the quality function Q 110:

$$Q (G; g, h) = \min_{a,b} \sum_{I=1}^{m} (G_i - \alpha g_i - \beta h_i)^2 / \sum_{I=1}^{m} G_i^2 \tag{3}$$

25   FIG. 1 shows a representation of the g 140 and h 150 vectors in 3 dimensional space. Each vector is defined by three components x, y and z, which are the expression levels of a feature over a set of three experiments. G, h and g are three arbitrary vectors in 3-dimensional space and the figure is for illustration purposes only and should not be interpreted as limiting the scope of the invention in any way. In 30 addition, the use of x, y, and z coordinate systems and the described planes should not in any way be interpreted as limiting the broad scope of the invention. The quality of a given pair g 140, h 150, as predictors of G 120, is the distance between the plane formed by these two vectors and target vector G 120 (See FIG. 1 for more details). G^P 125 is defined as the projection of G 120 on the plane 130 formed by g 140 and h 150.

Greedy Subset Search Algorithm (GSSA)

Referring now to FIGS. 1-3, the algorithm starts with an ordered, randomly selected subset of k-1 features. Next, this subset is complemented by one feature, which along with selected features forms the best possible (in quality) subset of k features. Number k is assigned to this feature. The feature number one is removed from the set and numbering of remaining genes: 2,.., k is reduced by one to become 1, ... k-1. This subset of k-1 features is an input to the next iteration of the loop 315 (See FIG. 3).

Iterations continue until the quality of a set of k predictors can not be further improved by replacing any one feature in the set 325 (See FIG. 3).

Method Outline: The Algorithm

Let $S = \{g_1 \ldots, g_n\}$ be expression levels of N features observed over the course of M experiments and let G 120 be an arbitrary target with expression levels over the same group of experiments. In this way, all features are represented by vectors in M-dimensional space. G 120 may or may not belong to S 200 (not shown in diagrams). In what follows, we will refer to every element of S 200 as a feature.

General Definition:

We say that a subset, $s \subset S$ 200, of features predicts the target G 120, with accuracy δ if the distance between the linear subspace generated by s and G 120 equals δ.

The Euclidean distance has been selected as a measure of proximity. Given this, and the linearity of the subspace, the definition above is simply stated as follows: the Least Squares distance between the subset s and the target G 120 equals:

$$\delta = \min \left\| G - \sum_{g_j \in S} \alpha_j g_j \right\| / \| G \| \qquad (4)$$

where min is taken over all sets of k real numbers α_j and $\| \; \|$ represents the norm in M-dimensional Euclidean space. The algorithm, in its current implementation, doesn't

take advantage of the specificity of the distance definition and, therefore, can be applied to any search problem of described nature with an arbitrary proximity measure. The value of δ will be referred to as the quality Q(s) 110 of a set of predictors s.

5   FIG. 2 shows a block diagram of the method of the present invention and how the algorithm works with the predictor and target data for a given size k of the predictor set. A target 120 of interest is first selected (not shown in the block diagram, but shown in FIG. 1). The method then selects an ordered subset of features of size k-1. If k=2, then the method randomly selects a feature, say g 140 (shown as reference

10   number 200 in the block diagram). To this subset the method now adds the complement feature, i.e., to form the best subset of size k (in terms of quality of prediction) that has the initially chosen subset of k-1 features. The quality of prediction of this subset of k features is then noted (shown as reference 220 in the block diagram). Next, the algorithm performs a checking step to see if the same set of

15   k features have appeared k times in a row (shown as reference number 225). If the answer is "yes" the algorithm stops and outputs the set of k features as the result. This is called an attractor (shown as reference numeral 230 in the diagram). If the answer is "no" the algorithm removes in order one feature from the predictor set (shown as 240 in the block diagram). The algorithm continues and repeats the steps of adding a

20   complement, checking the predictor set and removing a feature until the same set of k features has appeared k times in a row.

This process (of deleting a feature and adding another feature) may be repeated many times until a subset is reached whose quality of prediction can not be improved by deleting and then adding a single feature. This subset of size k (k=2

25   here) is referred to as an "attractor" 230 (shown generally as 230 is the block diagram of FIG. 2). The method then terminates and outputs the "attractor" as the best (in terms of quality of prediction of the target expression) subset of size k (k=2 here). The process can be modified, to probabilistically select the smallest (in size) subset of predictors that is closely related to the target in terms of the quality of prediction, as

30   follows. The method starts with an initial predictor set size of k=2 as described above. If the "attractor" set that the method outputs does not lie within the acceptable threshold of quality of prediction, the predictor set size is incremented by one, i.e., k=3. The algorithm uses the "attractor" of the previous stage (k=2) as the starting

11

subset of size k-1 at this stage. The method is iteratively applied with increasing value of k, until an "attractor" is found that is related to the target within the *a priori* set threshold of quality of prediction.

5   Referring now to FIG. 3, the definition of "attractor" and "complement" will now be clarified. FIG. 3 shows how a feature g 310 is first complemented by g* 320. g* 320 is a "complement" to g 310 implies that the set consisting of features g 310 and g* 320 has the best quality of prediction of all sets of size k=2 having feature g 310. Feature g 310 is then removed and a new "complement" feature $g^{(2)*}$ 330 is added to g* 320 and so on until the final predictor set $(g^{(k-1)*}, g^{(k)*})$ 325 is found, such

10   that the "complement" of $g^{(k)*}$ is the same as $g^{(k-1)*}$ (comparisons are shown by reference numeral 315). At this stage no deletion or addition of a single feature can improve the quality of prediction. Such a predictor set that can no longer be improved is called an "attractor".

15   **Definition of Complement(s):**

       Referring to FIG. 3, feature g* 320 is called a complement to a set, i.e. g*= (s)* if the quality of the set s ∪ g* is better or equal to the quality of s ∪ g for any choice of g 310, i.e.,

20

$$Q(s \cup g^*) \leq Q (s \cup g) \tag{5}$$

       Definition of Attractor(s):

25

       A set of features s is called an attractor (with respect to the complement) if (s')* ∪ s' = s for any subset s' of s that contains all but one element of s, i.e., one can not improve the quality of a set of predictors s by substituting only one feature (See FIG. 3).

30

The Problem and Algorithm:

For a given number k = 2,3...., a set of n features S, and a target G 120, to find the best quality subset of features s of size k. To solve the problem a random

5   algorithm has been designed. The main loop of this algorithm is defined as a cycle chosen for one random seed set s'. For a given k, the solution starts with the randomly selected seed set of genes s' of the size k-1.

One cycle of the main loop:

10

Let s [i] represent the ith element in s.

Set index = 0;

Randomly generate s' of size k-1;

Set current best attractor, s = ϕ (empty set);

15   Repeat unconditionally

{

Find the complement g* of s', i.e., g* = (s')*; /* there is always assumed seniority order in s, i.e., later complements have greater index, e.g., g* = s[k]*/

20

If (s'$\cup$ g* = s) then

{

s=s' $\cup$ g*;

index = index + 1;

25   if (index = k) then break out of the loop;

}

else /* find a new set of better quality than s*/

30   {

index = 1;

s = s' $\cup$ g*;

}

Form new seed set s' as s without its first element s[1];

}

Print set s as the predictor set;

5   The number of cycles in the main loop can be controlled by (i) the quality of a current set of predictors; (ii) total computational time; and (iii) direct constraint on the number of cycles itself. From a practical point of view, the above methods are suitable for parallel processing. Since each loop begins with a random subset of data, all that is necessary for the computation is to allow the method to generate these random

10 subsets, and compare with previously computed best subsets. Therefore, different initiations of the algorithm can be distributed across processors, as long as different random seed subsets are used in each loop and the computed best subsets are appropriately compared.

        The algorithm above has been provided so that one of ordinary skill in the art

15 can design and write code from it. Java software was used to code the algorithm in experimental runs. Experiments and data were run on a Personal Computer (PC) with a Windows NT (operating system). The system had 512MB RAM and a 800MHz CPU (processor). One of ordinary skill in the art needs to use a programming language to make a software prototype of the algorithm. He/she needs to have a

20 computer that has the software coding environment loaded. No special hardware requirements are necessary to run this algorithm. The larger the size of RAM and the more processing power (parallel processors would be even better), the better. Methods in exhaustive searches are also well known in the art and need not be described here in detail.

25

Algorithm performance:

        It can be shown that one cycle of the main loop always terminates and the predictor set it terminates at is an "attractor". It is obvious that the best (in quality) set

30 of genes must be an "attractor" as well. Therefore, given everything equal, the algorithm's performance depends on the number of "attractors"; the less the number of "attractors" the more likely the loop terminates at the best "attractor".

If π is the probability that one cycle of the main loop terminates at the best "attractor", then the probability that after P iterations of the main loop, the best "attractor" will be visited at least once is:

5

$$1 - (1 - \pi)^P$$

(6)

It has been observed in numerous computations using both generated and real data that as the number of "attractors" goes down when the number of experiments (dimensions of space) goes up.

10   In other words, as more independent experiments are conducted, the reliability of the data improves. Since the number of "attractors" reduces with increasing number of experiments, the probability of finding the best in quality (optimal) "attractor", using the GSSA algorithm, increases.

     The probability, π, of finding the optimal "attractor", however, goes down

15 with the increase in the number of genes, N, and the increase in the predictor set size, k. This is intuitive, since the number of possible "attractors" will increase with an increase in the number of genes or the size of the predictor set.

     The figures described and illustrated below were plotted using the software Matlab for PC. The PC was running Windows NT. Other methods well know in the

20 art can be used also to plot the data. Matlab was used because of its simplicity in requiring (x and y pairs) of data for plotting. These and other plotting methods are well known in the art. However, the described applications and plots should in no way be limiting of the broad scope of the invention. The program can be coded to run on other computers and different software may be used to plot the results. Figures 4-8

25 show plots of data from a series of runs on the GSSA algorithm. The plots show the application of the algorithm and method to gene reconstruction.

     FIG.4 shows a plot of (Nattr) plotted against Nexp. The plot shows the number of attractors of size 2 (i.e., k=2) as dimensionality of the data increases. Each gene is represented as a vector in m-dimensions. These m-dimensions are the m experiments

30 that were conducted on all the genes in the set S. This simulation consists of expression of 200 genes in 40 experiments. Thus, size of the set S is 200, and m=40. The graph shows how the number of attractors of size 2 (k=2) decreases as more experimental values are considered. It shows that as we use more experiments, we

have less number of attractors and hence, a better chance at reaching the best solution in a few random runs of the GSSA algorithm. The use of the algorithm provides a special blessing of dimensionality toward a final solution. As has been described above, the number of "attractors" can actually be controlled by the number of

5   experiments. In addition, the number of "attractors" can be estimated *a priori* (proof beyond the scope of the invention). Existence of multiple "attractors" for a given data set may be evidence of insufficient amount of experiments to draw confident conclusions regarding the nature of dependencies among genes. It also suggests a specific number of additional experiments to be performed to substantiate the

10 conclusion.

Another point to be noted is that there is a tradeoff between the number of genes replaced at each step of the algorithm and the quality of "attractor". For instance, the computational time increases exponentially as the number of replaced genes goes up, but the algorithm may stop at a better quality "attractor". However,

15 one gene replacement algorithms converge to a high "quality" attractor with high probability, given a sufficient number of experiments.

FIG. 5 shows a simulated plot of the probability of finding the optimal predictor set in a single run of the approximate algorithm. As can be determined from the diagram, as the number of genes increases the probability in most cases decreases

20 in finding the actual predictor set. Fig 5. shows the probability to find the best solution (best attractor of size k=numP in the graph) from a set of N (x-axis of the graph) genes in one run of GSSA. One run of GSSA means, starting with one random seed and running the GSSA algorithm until an attractor is found. The dimensionality of the data here (the number of experiments or m was fixed and was 38). Three plots

25 are shown in this figure. The top plot shows the results for k=numP=2. The middle plot shows results for k=numP=3 and the bottom plot shows results for k=numP=4. The top plot and other plots are similar. The top plot (k=numP=2) shows that the probability of finding the best solution drops as we increase the number of genes. This is because there exist more attractors as the number of genes increases. Now the

30 difference in the three plots also shows that as we keep the number of genes fixed (take a fixed value on the x-axis), but increase the value of k (i.e., increase k from 2 to 4), we see that the probability of finding the best solution also decreases (and this too is because the number of attractors increases as the size of k increases). If π is the

16

probability of reaching the best solution in one run of GSSA, then the probability that it reaches the best solution in p trials is $1-(1-\pi)^p$ which can be brought close to 1 if we increase p.

FIG. 6 shows a simulated plot that demonstrates the computational complexity of the approximate algorithm increases linearly as opposed to an exponential increase using an exhaustive search. The diagram shows a plot of execution time vs. number of genes. It becomes evident that as the number of genes increases, the execution time increases in a linear fashion. The fact that the execution times (as the number of genes increases) lie on a line indicates the linear nature of the algorithm. Fig 6. shows the execution time of running GSSA 50 times versus the total number of genes. We take 50 random start seeds and run the GSSA until it reaches an attractor for each of the 50 instances. Then we select as the best answer, the best result of the 50. The size of set S of genes from which to choose the predictors is increased in the experiment. We see that the execution time increases linearly. The three plots show for three cases (k=numP=2, 3, 4). The exhaustive search that finds the best solution has an exponential increase in time.

FIG. 7 shows a simulated plot of the execution time of the algorithm as a function of the number of predictors. The results of the plot indicate that as the predictor set size increases, the execution time also increases, but in a linear fashion. In other words, the execution time does not increase at the same exponential rate as exhaustive search when there is an increase in the number of predictors. This is very important for calculations necessary in the gene network reconstruction. What may take years to complete (due to the exponential nature of exhaustive search) may now be completed in a matter of few minutes. Fig 7. keeps number of genes fixed (S is fixed) and varies k=numP and plots the execution time for 50 runs of the GSSA algorithm. The execution times can be divided by 50 to yield time to run for a single run of GSSA (take one random seed and run GSSA until you reach an attractor). Again the execution time increases linearly with increase in k. Two plots are shown for size of S = 10, and 20. Note that the execution times for Fig 6. and Fig 7. are for the algorithm implemented in Java and running on a PC running Windows NT with a CPU of 800 MHZ and 256MB RAM(memory).

FIG. 8 shows a plot of the log of execution time against number of genes. The trend in this plot is similar to the trends presented in the previous plots. The plot

17

clearly shows the novelty and power of the present algorithm in finding "good" solutions that may be effective in the gene network reconstruction problem. Fig 8. also shows the difference between execution times for the GSSA algorithm and the exhaustive search. The number of genes (size of S) is varied and the execution times

5   are shown as log (to the base e – natural logarithm). Here, we show time for only 1 run of GSSA (as against 50 runs of the algorithm in Fig 6. and 7.). This shows that this algorithm is extremely fast as compared to exhaustive search methods that are well known in the art.

10   **Further Applications of the Invention**

        The method of invention can easily be extended to probabilistically determine the smallest (in size) subset of features that are closely related to a given target within an *a priori* set tolerance level in terms of a selected quality function. The method then

15   takes as input a given target (user selected) and iteratively applies the method of invention for subsets of size 1, 2, 3, ..., k, etc., until a predictor set that is closely related to the target expression, within the *a priori* set threshold, is found.

        The method of the invention can also be used for classification. The method now defines a target in terms of a vector of numbers representing the class of the

20   experiments under consideration. The method of invention can be used to identify a subset of features that can predict the class of the experiment within the given quality of prediction. As an example (though not limiting the use of the invention), we can consider a micro-array data for say, leukemia. The data consists of gene expression results for different tissues (a tissue sample represents one experiment) representing

25   various types of leukemia. If we assign a number for each type of leukemia (say, 0, 1, 2, etc.), then we can define a target over all the experiments by the vector of numbers representing the type of leukemia represented by the tissues. We can now use the method of invention to identify a "good" subset of genes that can predict the target vector (and hence the type of tissues). Hence, these genes can form a diagnostic set of

30   genes whose expression values are used to discriminate between the various types of leukemia.

        Various modifications to the embodiments of the invention described above are, of course, possible. Accordingly, the present invention is not limited to the particular embodiments described in detail above.

                                          18

References

Bishop, C.M., "Neural Networks for Pattern Recognition", Clarendon Press, Oxford,
5   1997.

M. Bittner et al., "Molecular Classification of Cutaneous Malignant Melanoma by
Gene Expression Profiling", Nature, Vol. 406, August 2000.

10   P.Pudil, J.Novovicova, J.Kittler, "Floating Search Methods in Feature Selection",
Pattern Recognition Letters 15 (1994) 1119-1125, Elsevier Science B.V..

Somol P., Pudil P., "Oscillating Search Algorithms for Feature Selection",
Proceedings 15th International Conference on Pattern Recognition, Vol. 2, 406-409,
15   Barcelona, Spain, 2000.

http://www-binf.bio.uu.nl/~dutilh/gene-networks/thesis.html.

20

25

30